

## **IN THE CLAIMS**

Please replace the claims as follows:

1. (original) A computer-implemented method for optimization of an executable program that calls procedures in a shared library, comprising:

identifying linkage code segments in the executable program, wherein each linkage code segment is associated with a call to a procedure in the shared library, reads procedure address information from a linkage table, and transfers control to an associated one of the procedures;

reading the address information from the linkage table;

generating respective substitute code segments for the linkage code segments, each substitute code segment having references to the linkage table replaced by direct loads of the address information without reference to the linkage table; and

executing the substitute code segments instead of the linkage code segments.

2. (original) The method of claim 1, further comprising:

allocating relocation address space for the executable program;

storing the substitute code segments in the relocation address space.

3. (original) The method of claim 1, further comprising for each calling procedure that calls a called procedure in the shared library:

identifying registers that are not used by the calling procedure;

generating in a substitute code segment instructions that store in the registers the address information read from the linkage table.

4. (original) The method of claim 1, further comprising:

annotating linkage code segments in the executable program during compilation;

placing breakpoints at locations of the linkage code segments as indicated by the annotations; and

upon encountering a breakpoint at a linkage code segment, generating a substitute linkage code segment.

5. (original) The method of claim 4, further comprising:

identifying in the annotations registers that are not used by a calling procedure that calls a called procedure in the shared library; and

generating in a substitute code segment instructions that store in the registers indicated by the annotations the address information read from the linkage table .

6. (original) The method of claim 1, wherein the executable program includes a calling procedure that calls a first procedure and a second procedure in a first shared library and calls a third procedure in a second shared library, further comprising:

identifying a set of registers that are not used by the calling procedure;

generating in a substitute code segment instructions that store in the registers in the set the address information read from the linkage table

configuring a substitute code segment associated with the first procedure to load the address information from one of the registers in the set into a branch register prior to transferring control to the first procedure; and

eliminating from a substitute code segment associated with the second procedure, an instruction that loads the address information into the one of the registers if the second procedure is called after the first procedure and before the third procedure.

7. (original) The method of claim 1, wherein the executable program includes a call to a first procedure in a first shared library that calls a second procedure in a second shared library, further comprising in the substitute code segment associated with the call by the first procedure to the second procedure, replacing an indirect branch instruction that targets the second procedure with an instruction-pointer relative branch instruction that targets the second procedure.

8. (original) The method of claim 1, further comprising:

placing breakpoints at procedure entry points;

upon encountering a breakpoint at the entry point of a procedure during program execution, searching for linkage code segments in the procedure, generating substitute linkage code segments for those identified in the procedure, and continuing with execution of the procedure.

9. (original) The method of claim 8, further comprising:

allocating relocation address space for the executable program;  
storing the substitute code segments in the relocation address space.

10. (original) The method of claim 8, further comprising for each calling procedure that calls a called procedure in the shared library:

identifying registers that are not used by the calling procedure;  
generating in a substitute code segment instructions that store in the registers the address information read from the linkage table.

11. (original) The method of claim 8, wherein the executable program includes a call to a first procedure in a first shared library that calls a second procedure in a second shared library, further comprising in the substitute code segment associated with the call by the first procedure to the second procedure, replacing an indirect branch instruction that targets the second procedure with an instruction-pointer relative branch instruction that targets the second procedure.

12. (original) An apparatus for optimization of an executable program that calls procedures in a shared library, comprising:

means for identifying linkage code segments in the executable program, wherein each linkage code segment is associated with a call to a procedure in the shared library, reads procedure address information from a linkage table, and transfers control to an associated one of the procedures;

means for reading the address information from the linkage table;

means for generating respective substitute code segments for the linkage code segments, each substitute code segment having references to the linkage table replaced by direct loads of the address information without reference to the linkage table; and

means for executing the substitute code segments instead of the linkage code segments.

13. (new) A computer-implemented method for optimization of an executable program, comprising:

during execution of a program that calls procedures in a shared library,

identifying linkage code segments in the executable program, wherein each linkage code segment is associated with a call to a procedure in the shared library, reads procedure address information from a linkage table, and transfers control to an associated one of the procedures;

reading the address information from the linkage table;

generating respective substitute code segments for the linkage code segments, each substitute code segment being a copy of a linkage code segment with references to the linkage table in the linkage code segment replaced by direct loads of the address information without reference to the linkage table; and

executing the substitute code segments instead of the linkage code segments, wherein each linkage code segment transfers control to a procedure in the shared library without referencing the linkage table.

14. (new) The method of claim 13, further comprising:

during execution of the program that calls procedures in a shared library,

allocating a first address space for the program and additional address space for substitute code segments; and

writing the substitute code segments to the additional address space.

15. (new) The method of claim 14, further comprising:

during execution of the program that calls procedures in a shared library,

in response to a call to a first procedure in the shared library made within a second procedure in the program and a linkage code segment that references the linkage table for addressing the first procedure, copying the second procedure from the first address space to the additional address space; and

overwriting a first instruction in the second procedure in the first address space, with branch instruction that branches to the second procedure in the additional address space.

16. (new) The method of claim 13, further comprising:

during execution of the program that calls procedures in a shared library,

identifying each calling procedure that calls a called procedure in the shared library;

identifying registers that are not used by the calling procedure; and  
generating in a substitute code segment instructions that store in the registers  
the address information read from the linkage table by a linkage code segment.

17. (new) An article of manufacture, comprising:

a processor-readable medium configured with instructions that when executed cause a  
processor, during execution of a program that calls procedures in a shared library, to,

identify linkage code segments in the executable program, wherein each  
linkage code segment is associated with a call to a procedure in the shared library,  
reads procedure address information from a linkage table, and transfers control to an  
associated one of the procedures;

read the address information from the linkage table;

generate respective substitute code segments for the linkage code segments,  
each substitute code segment being a copy of a linkage code segment with references  
to the linkage table in the linkage code segment replaced by direct loads of the address  
information without reference to the linkage table; and

execute the substitute code segments instead of the linkage code segments,  
wherein each linkage code segment transfers control to a procedure in the shared  
library without referencing the linkage table.

18. (new) The article of manufacture of claim 17, further comprising:

instructions on the medium that when executed cause the processor, during execution of the  
program that calls procedures in a shared library, to

allocate a first address space for the program and additional address space for  
substitute code segments; and

write the substitute code segments to the additional address space.

19. (new) The article of manufacture of claim 18, further comprising:

instructions on the medium that when executed cause the processor, during execution of the  
program that calls procedures in a shared library, to

in response to a call to a first procedure in the shared library made within a  
second procedure in the program and a linkage code segment that references the

linkage table for addressing the first procedure, copy the second procedure from the first address space to the additional address space; and

    overwrite a first instruction in the second procedure in the first address space, with branch instruction that branches to the second procedure in the additional address space.

20. (new) The article of manufacture of claim 17, further comprising:

instructions on the medium that when executed cause the processor, during execution of the program that calls procedures in a shared library, to

    identify each calling procedure that calls a called procedure in the shared library;

    identify registers that are not used by the calling procedure; and

    generate in a substitute code segment instructions that store in the registers the address information read from the linkage table by a linkage code segment.